

Technical Disclosure Commons

Defensive Publications Series

October 15, 2018

Data Saving Using Contextual Signals

Victor Carbune

Thierry Coppey

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Carbune, Victor and Coppey, Thierry, "Data Saving Using Contextual Signals", Technical Disclosure Commons, (October 15, 2018)
https://www.tdcommons.org/dpubs_series/1592



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Data saving using contextual signals

ABSTRACT

Mobile device applications and operating systems offer a data-saving option such that online content that is believed to be unchanged since last load is not reloaded. However, this sometimes leads to a situation where online content that has actually changed is not reloaded or that stale content is reloaded. The misdetection of fresh content as stale or vice-versa occurs due to the heuristics that are used by the data-saving algorithms of the application/ OS, due to misconfigured servers, etc.

This disclosure presents machine-learning techniques that determine if a web page or a portion thereof is to be reloaded. The techniques use various contextual signals, as permitted by the user, to make the reload decision, e.g., the content to be reloaded; surrounding content; their metadata and positions on a web-page; user interaction and behavior with the website; previously loaded content; etc. The techniques enable data-saving techniques that are robust and tailored to both user and website.

KEYWORDS

- mobile data
- page reload
- page refresh
- stale content
- smart refresh
- data saving
- web browser

BACKGROUND

Various mobile apps, browsers, and operating systems offer a data-saving option such that online content, e.g., images, videos, text, etc., that is believed to be unchanged since last load is kept identical and not reloaded, e.g., when a user visits a web page. However, this sometimes leads to a situation where online content that has actually changed is not reloaded or that stale content is reloaded. The misdetection of fresh content as stale or vice-versa occurs due to the heuristics that are used by the data-saving algorithms of the application/OS; misconfigured servers; overly aggressive data-saving setting combined with dynamic content on websites; etc.

DESCRIPTION

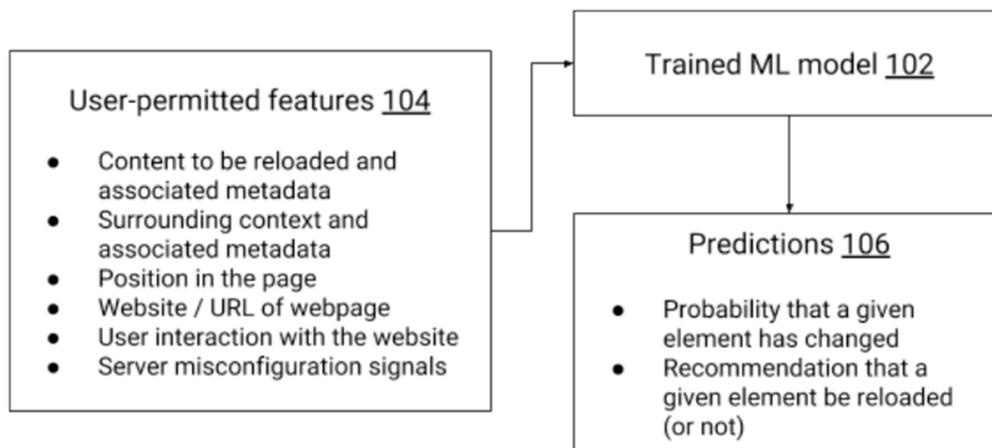


Fig. 1: Applying machine learning model to recommend reloads

Per techniques of this disclosure (as illustrated in Fig. 1), a trained machine learning model (102) is utilized to make predictive recommendations (106) regarding the fresh/stale state of a webpage or element thereof. The predictions are based on a number of features (104) that are permitted by the user for use in determining the predictive recommendations. Only such

features as permitted by the user are utilized. Some of the features (or signals) that are used to make predictions are:

- Content to be reloaded and its metadata: Some content (e.g., sports items, fast-changing news) are likely to change fast, e.g., every few seconds, every few minutes, etc., and others (e.g., blog posts) may change slowly, e.g., after a few hours, after a few days, etc. The machine learning model is trained and learns the rate of change of various online content via training. The metadata of the content includes type, data last loaded, date on server if available, immediate previous version in cache if available, etc.
- Surrounding context and their metadata: As an example, if an image is being considered for reload, the surrounding context (e.g., within a webpage) may be text describing the image or in proximity to it. The surrounding context, previously loaded content, their fresh/stale state, their metadata, etc. are useful signals that are utilized to predict if a given piece of content is to be reloaded.
- Position of the content on the web-page: A centrally located element, e.g., image, is often worthwhile reloading, while peripheral elements (side-bar or menu-item thumbnails) on a web page are generally slower to change. Thus, the position of the content with a page is a useful signal that is utilized by the machine-learning model.
- URL of the server or website: Some websites, e.g., news or entertainment websites, have highly dynamic content. Others, e.g., literary or educative websites, change more slowly. The URL of the server or website is utilized as by the machine-learning model as a signal to make a reload/ don't reload recommendation.
- User interaction with website: Certain users may frequently force refresh of a web page, e.g., to be up-to-date on a sports match or breaking news. Other users may absorb content

from the same website at a more leisurely pace. Thus, respective user interactions with a website, e.g., intervals between forced refresh, if permitted by the user are utilized by the machine-learning model.

- Server misconfiguration signals: The server that transfers content via http protocol may send out a date of expiration for a given content. If the server doesn't do so, or sends an incorrect expiration date, or indicates that a certain nearly-static element shouldn't be cached, then it is likely that the server is misconfigured. A client that follows server instructions and keeps reloading content only to find it unchanged learns, per the techniques herein, to ignore the misconfigured server. Further, a misconfigured server might indicate no reload even though the content did actually change. This is another kind of server misconfiguration that is detected by the techniques of the disclosure. Server misconfiguration is sometimes indicated by user behavior. For example, if the user forces refresh, such refresh it is an indication that the user perceives at least some of the on-screen content to be stale, even if the server indicates that the content is fresh. Additionally, the machine-learning model can be initialized with a list of known misconfigured servers, and configured to adjust the list over time.

The predictions provided by the machine learning model can include a probability score that a given element has changed; a recommendation that a given element be reloaded (or not); etc.

The machine learning model can be, for example, a standard binary classification model or a regression model if probability or score values are available in the training data, etc. The model can be implemented using neural networks, including long short-term memory (LSTM) neural networks, word embeddings, recurrent neural networks, convolutional neural networks, etc. Other machine learning models, e.g., support vector machines, random forests, boosted

decision trees, two-arm contextual bandit models, etc., can also be used. Reinforcement learning can be used to advantageously incorporate past predictions as feedback to improve machine-learning performance. Feedback to the model can include the results of past decisions, e.g., whether a prediction of changed content was accurate. The model is trained with known pairs of input features and predictions. The machine learning can be pre-trained, e.g., using a model that has trained on training data from a variety of clients, and over time, the model gets specific to the user if the user permits use of user-specific data, e.g., to update a local model implemented on the user device.

For the scenario when the server is misconfigured, an alternate (or additional) approach can include determining if the server itself (e.g., IP/domain) is misconfigured, e.g., if it is signaling do-not-cache messages. This can be done by checking (e.g., per extension) which elements have changed, and by checking which elements are not cached but still do not have an expiration policy.

In this manner, an improved data-saving user experience can be achieved by incorporating user-permitted contextual signals, and by training an adaptive machine learning model that produces a likelihood that a given content has changed. When a user permits, the techniques are tuned to the user and the website, e.g., if certain sections of content in a website are found to be uninteresting to the particular user over multiple uses, then those sections may not be refreshed at all. The model can be combined with other client-side data-saving heuristics. In comparison to traditional server-administrator or web-developer directed data saving, the techniques of this disclosure are more naturally aligned to the interests of the end-user, e.g., efficiency, speed of download, cost of download, etc. Users have more fine-grained control in the amount of data they consume.

In situations in which certain implementations discussed herein may collect or use personal information about users (e.g., user data, information about a user's social network, user's location and time at the location, user's biometric information, user's activities and demographic information), users are provided with one or more opportunities to control whether information is collected, whether the personal information is stored, whether the personal information is used, and how the information is collected about the user, stored and used. That is, the systems and methods discussed herein collect, store and/or use user personal information specifically upon receiving explicit authorization from the relevant users to do so.

For example, a user is provided with control over whether programs or features collect user information about that particular user or other users relevant to the program or feature. Each user for which personal information is to be collected is presented with one or more options to allow control over the information collection relevant to that user, to provide permission or authorization as to whether the information is collected and as to which portions of the information are to be collected. For example, users can be provided with one or more such control options over a communication network. In addition, certain data may be treated in one or more ways before it is stored or used so that personally identifiable information is removed. As one example, a user's identity may be treated so that no personally identifiable information can be determined. As another example, a user's geographic location may be generalized to a larger region so that the user's particular location cannot be determined.

CONCLUSION

This disclosure presents machine-learning techniques that determine if a web page or a portion thereof is to be reloaded. The techniques use various contextual signals, as permitted by the user, to make the reload decision, e.g., the content to be reloaded; surrounding content; their

metadata and positions on a web-page; user interaction and behavior with the website; previously loaded content; etc. The techniques enable data-saving techniques that are robust and tailored to both user and website.